R Programming

Prof. Ramesh Ragala

VIT University Chennai Campus

July 10, 2017

Introduction

- R is an integrate suite of software facilities of

Introduction

- R is an integrate suite of software facilities of
  1. **Data Manipulation**

Introduction

- R is an integrate suite of software facilities of
  1. **Data Manipulation**
     - Effective data handling and storage facility

Introduction

- R is an integrate suite of software facilities of
  1. **Data Manipulation**
     - Effective data handling and storage facility
  2. **Calculations**

Introduction

- R is an integrate suite of software facilities of
  1. **Data Manipulation**
     - Effective data handling and storage facility
  2. **Calculations**
     - Various operators for calculations on arrays/matrices
     - Well-developed language including conditionals, loops, recursive functions and I/O capabilities

Introduction

- R is an integrate suite of software facilities of
  1. **Data Manipulation**
     - Effective data handling and storage facility
  2. **Calculations**
     - Various operators for calculations on arrays/matrices
     - Well-developed language including conditionals, loops, recursive functions and I/O capabilities
  3. **Graphical Display**

Introduction

- R is an integrate suite of software facilities of
  1. **Data Manipulation**
     - Effective data handling and storage facility
  2. **Calculations**
     - Various operators for calculations on arrays/matrices
     - Well-developed language including conditionals, loops, recursive functions and I/O capabilities
  3. **Graphical Display**
     - Graphical facilities for data analysis

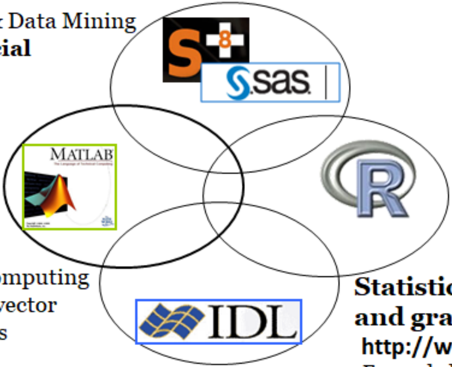
Introduction

- R is an integrate suite of software facilities of
  1. **Data Manipulation**
     - Effective data handling and storage facility
  2. **Calculations**
     - Various operators for calculations on arrays/matrices
     - Well-developed language including conditionals, loops, recursive functions and I/O capabilities
  3. **Graphical Display**
     - Graphical facilities for data analysis
  4. **Statistical Computing**

Introduction

- R is an integrate suite of software facilities of
  1. **Data Manipulation**
     - Effective data handling and storage facility
  2. **Calculations**
     - Various operators for calculations on arrays/matrices
     - Well-developed language including conditionals, loops, recursive functions and I/O capabilities
  3. **Graphical Display**
     - Graphical facilities for data analysis
  4. **Statistical Computing**
     - It supports linear and non-linear modelling, classical statistical tests, etc

Introduction



- Statistics & Data Mining
- **Commercial**

- Technical computing
- Matrix and vector formulations

- Data Visualization and analysis platform
- Image processing, vector computing

Statistical computing and graphics

http://www.r-project.org

- Expanded by community as **open source**
- Statistically rich

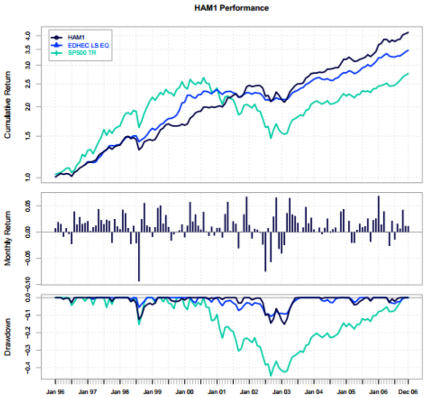
Introduction

- R is based on the S language originally developed by John Chambers and colleagues at AT&T Bell Labs in the late 1970s and early 1980s
- R (sometimes called "GNU S") is free open source software licensed under the GNU general public license (GPL 2)
- R development was initiated by **Robert Gentleman** and **Ross Ihaka** at the University of Auckland, New Zealand in the 1990s
- R is formally known as The R Project for Statistical Computing
    - **www.r-project.org**

Introduction

- **Strengths of R Programming Language:**

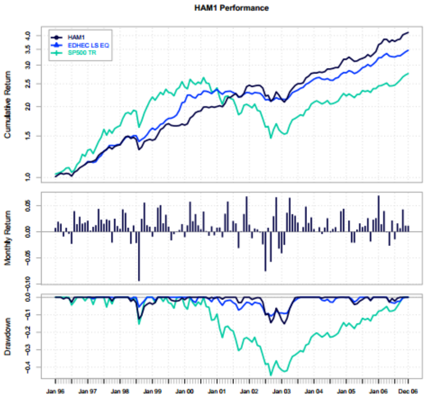
1. **Data Manipulation**



Introduction

- **Strengths of R Programming Language:**

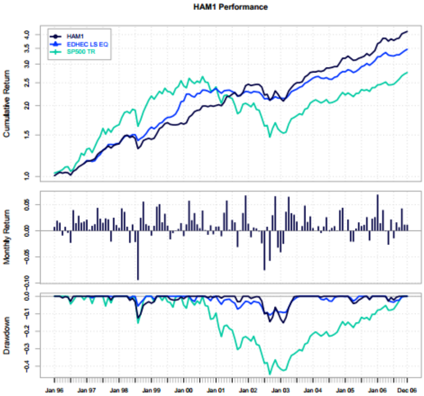
1. **Data Manipulation**
2. **Data Analysis**



Introduction

- **Strengths of R Programming Language:**

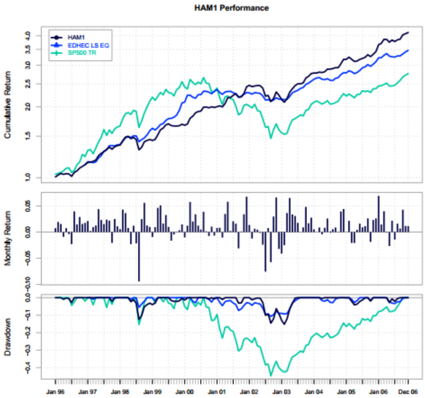
1. **Data Manipulation**
2. **Data Analysis**
3. **Statistical Modeling**



Introduction

- **Strengths of R Programming Language:**

1. **Data Manipulation**
2. **Data Analysis**
3. **Statistical Modeling**
4. **Data Visualization**



Introduction

- **S Language**

- Original S → AT & T Bell Labs
- S-PLUS → S + GUI
- R → The R Project for Statistical Computing
- R is the most recent and full-featured implementation of the S language

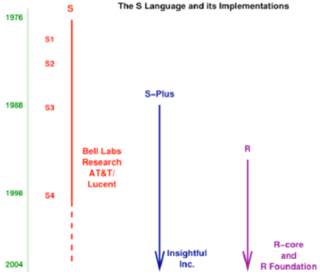
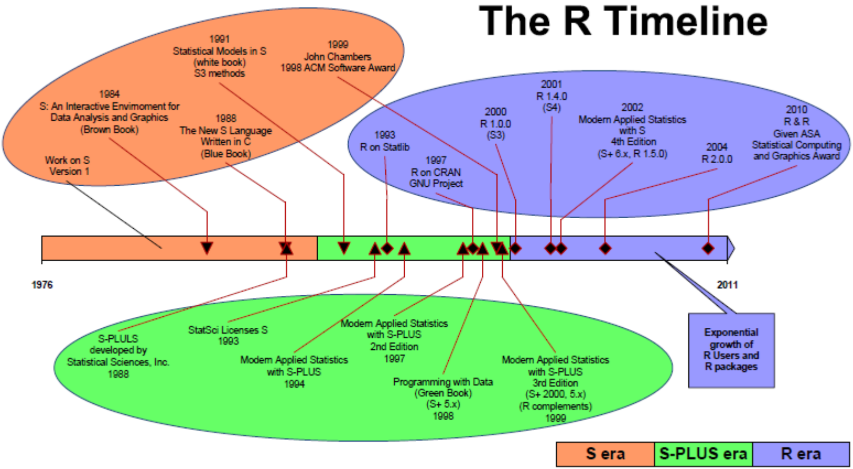


Figure from *The History of S and R, John Chambers, 2006*

Introduction



The R Timeline

1976

2011

Work on S Version 1

1984 S: An Interactive Envirnoment for Data Analysis and Graphics (Brown Book)

1988 The New S Language Written in C (Blue Book)

1991 Statistical Models in S (white book) S3 methods

1999 John Chambers 1998 ACM Software Award

1993 R on Statlib

1997 R on CRAN GNU Project

2000 R 1.0.0 (S3)

2001 R 1.4.0 (S4)

2002 Modern Applied Statistics with S 4th Edition (S+ 6.x, R 1.5.0)

2004 R 2.0.0

2010 R & R Given ASA Statistical Computing and Graphics Award

S-PLULS developed by Statistical Sciences, Inc. 1988

StatSci Licenses S 1993

Modern Applied Statistics with S-PLUS 1994

Modern Applied Statistics with S-PLUS 2nd Edition 1997

Programming with Data (Green Book) (S+ 5.x) 1998

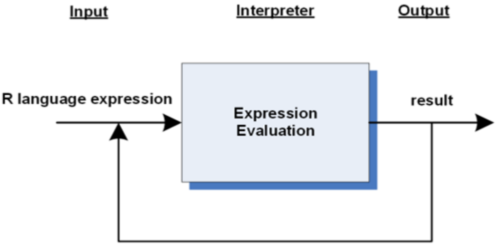
Modern Applied Statistics with S-PLUS 3rd Edition (S+ 2000, 5.x) (R complements) 1999

Exponential growth of R Users and R packages

S era	S-PLUS era	R era

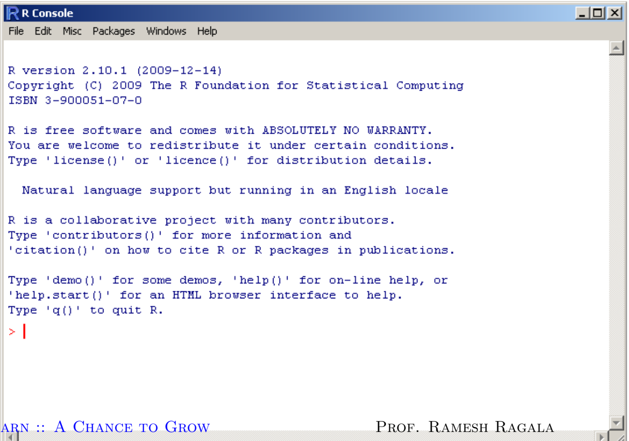
Introduction

- R is an interpreted language
- R Interpreter → evaluate R-Commands or execute R-scripts
  - **Rgui**
  - **RStudio**
- R- Expression evaluation
  - R Expressions are processed via R's **Read-Evaluate-Print-Loop** → REPL



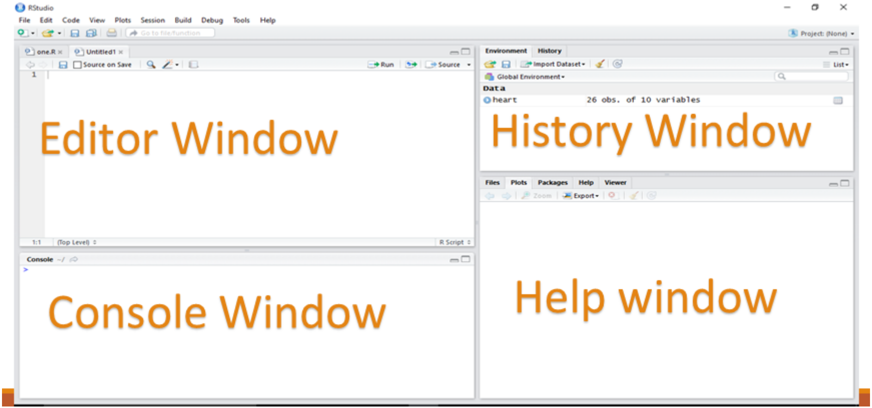
Introduction

- RGui
  - RGui is an interactive command driven environment.
  - we can type R – Commands (including expressions) in R Console.
  - We are able to copy and paste multiple commands in R Console



```
R version 2.10.1 (2009-12-14)
Copyright (C) 2009 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```

- RStudio Layout



```
Data
heart          26 obs. of 10 variables
```

Introduction

- **Console Window**
  - It is situated at Bottom Left of the RStudio Layout
  - It is also called command window
  - We can write commands
  - All commands can be executed in this window only
- **Editor Window**
  - It is situated at Top Left of corner of the RStudio Layout
  - It is used for writing scripts → collection of commands
  - It is also called script window
  - If this window is not visible, we can get it by **File** → **New** → **Rscript**
  - Click **RUN** or **CRTL+ENTER** to send the highlighted commands to command window

Introduction

- **History Window**
  - It is situated at Top Right of corner of the RStudio Layout
  - In this window, we can see the data and values of R which are currently stored in memory.
  - It is also called workspace window
  - We can view and edit the values by clicking
  - This history window shows what has been typed so far or objects created so far.
- **Help Window**
  - It is situated at the right bottom of RStudio Layout
  - Here we can open files and view plots
  - We can **install** and load the packages

Introduction

- **Short-cut Keys in RStudio**
  - Easy running of the code → CTRL+ENTER (runs highlighted lines of code)
  - Even easier → CTRL+ENTER+P re-run the last-run code
  - CTRL+1 → source editor
  - CTRL+2 → Console
  - CTRL+L → clear the console
  - CTRL+O → Open the file
  - CTRL+S → save the fie
  - CTRL+shift+N → opens new document
  - **ESC → Interrupt a lengthy R command**
  - CTRL+shift+C → comment or un-comment (highlighted code)

Introduction

- **R - Help System**
  - R has Comprehensive HTML help facility.
  - **help.start()** in R-Console window → it results the R-language documentation in HTML Pages
  - Displays the Help file for a specific function
    - **help(”topicname”)**
    - **?topic**
  - help.search() → Searches for a word in the Help files
    - **help.search()**
    - **??topic**

```
help.start()
help(”data.frame”)
help(data.frame)
?data.frame
??data.frame
help.search(”predict”)
??”predict”
```

Introduction

- **Setting the Working Directory**
  - To store working file
  - Create a folder with name as **RdataWork**
  - **setwd(new.dir.path)** → changes the working directory
  - **getwd()** → Returns the current directory

```
setwd("E:/VIT/RdataWork")
getwd()
```

Introduction

- **R as a Simple Calculator**
  - Typing in a mathematical expression and hitting enter prints the result.
  - Order of operation rules worked as expected
  - Mathematical functions are also supported
  - The result of mathematical expression can be assigned to an object in R
  - Every object in R belongs to a class → The type of the object it represents
  - Everything in R is an object, including functions.
  - **ls()** → prints all objects

```r
10+20
sqrt(36)
var1 <- sqrt(81)
var1
class(var1)
ls()
```

Introduction

- **R as a Simple Calculator**
  - Few maths functions are **abs**, **sqrt**, **log**, **exp**, **log10**, **factorial**, etc.
  - Few Trig functions are **sin**, **cos**, **tan**, **asin**, **acos**, **atan**, etc.
  - Few Rounding functions are **round**, **ceiling**, **floor**, **trunc**, **signif**, **zapsmall**, etc.
  - Few math quantity functions are **Inf**, **-Inf**, **NaN**, **pi**, **exp(1)**, **1i**, etc

```r
5%%4 # modulo operator
log(2)
cos(pi)
sin(0)
asin(0)
ceiling(3.2)
0/0 # it produce NaN -> Not a Number
1/Inf
factorial(5)
```

Introduction

- **R as a Simple Calculator**

```r
round(123.456,digits = 2)
round(-123.456, digits = 2)
round(-123.456, digits = -2)
signif(-123.456, digits = 4) # number of significants digits to be retained
# floor(x) rounds to the nearest integer that's smaller than x
floor(123.45)
floor(-123.45)
# trunc(x) rounds to the nearest integer in the direction of 0.
trunc(123.65)
trunc(-123.65)
```

Introduction

- **Basic Data types in R**
  - **Numeric**
  - **Integer**
  - **Complex**
  - **Logical**
  - **Character**
  - **Vector**
  - **Matrix**
  - **List**
  - **Data Frame**

Introduction

- **Numeric in R**
    -

Introduction

- **Numeric in R**

```r
var2 <- 25.12
var2
print(var2)
# is.integer() -> used to check whether a given variable object is integer or not
is.integer(var2)
is.integer(25)
typeof(var2)
typeof(25)
```

Introduction

- **Integers in R**
  - **as.integer()** → used to create integer value

```r
var3 <- as.integer(99)
var3
is.integer(var3)
typeof(var3)
class(var3)
var4 <- as.integer(123.45)
var4
is.integer(var4)
typeof(var4)
class(var4)
```

Introduction

- **Logical Values**
  - **Logical AND (&)**, **Logical OR (|)** and **Negation (!)**

```
as.integer(TRUE)
as.integer(FALSE)
x = 1
y = 2
z = x > y
z
class(z)
k= x & y
k
class(k)
m=0
n=x & m
n
class(n)
f = !m
f
class(f)
```

Introduction

- **Complex Values**
  - A complex value in R is defined via the pure imaginary value i.

```
as.integer(TRUE)
as.integer(FALSE)
x = 1
y = 2
z = x > y
z
class(z)
k= x & y
k
class(k)
m=0
n=x & m
n
class(n)
f = !m
f
class(f)
```

THANK YOU!